FEATURE DESIGN USING AUDIO DECOMPOSITION FOR INTELLIGENT CONTROL OF THE DYNAMIC RANGE COMPRESSOR

Di Sheng, György Fazekas

Centre for Digital Music (C4DM), Queen Mary University of London

ABSTRACT

This paper proposes a method of controlling the dynamic range compressor using sound examples. Our earlier work showed the effectiveness of random forest regression to map acoustic features to effect control parameters [1]. We extend this work to address the challenging task of extracting relevant features when audio events overlap. We assess different audio decomposition approaches such as onset event detection, NMF, and transient/stationary audio separation using ISTA and compare feature extraction strategies for each case. Numerical and perceptual similarity tests show the utility of audio decomposition as well as specific features in the prediction of dynamic range compressor parameters.

Index Terms— Audio signal processing; Intelligent production; Dynamic range compressor; Audio decomposition.

1. INTRODUCTION

Despite the prevalence of computers and Digital Audio Workstations (DAW) in music production, most audio engineering tasks remain labour intensive and reliant on hard-to-acquire skills. Musicians trying to produce their own tracks for instance often lack experience in configuring audio effects. This involves tweaking low-level signal processing parameters given an aesthetic goal concerning some desired sound qualities. The parameters however have limited meaning from a musical perspective. Intelligent control tools of audio effects therefore have the potential to democratise music production, facilitate the learning process for beginners or enable professionals to concentrate on aesthetic choices rather than technical decisions.

This paper focuses on the use of sound examples to control the dynamic range compressor (DRC), an essential effect in many audio production use cases. This modality received little attention in the significant body of work on intelligent audio production [2]. In our work, a set of acoustic features are used to capture important characteristics of sound examples. These are then mapped to audio effect control parameters using regression. We propose different audio decomposition and feature extraction strategies to analyse and process mono-timbral audio loops. The rest of the paper is organised as follows. Section 2 provides the essential background. Details about the decomposition methods are given in Section 3. Evaluation results are reported in Section 4, followed by conclusion and future work outlined in Section 5.

This work was part funded by the European Commission H2020 research and innovation grant AudioCommons (688382).

2. BACKGROUND

Automatic control of audio effects has become an important topic in the fields of intelligent music production and automatic mixing over the last decade. A thorough review is provided in [2], therefore we only mention a few pertinent works here. Creating a technically correct mix through controlling loudness balance or dynamic range of sources has become a common task. For instance, the authors in [3] aim at finding the optimal dynamic range for each track considering domain knowledge gathered from audio engineers. Cartwright et. al. [4] outline a control strategy using high-level semantic terms such as *warm* or *harsh*, while research presented in [5] targets new graphical interfaces.

Our work is different from previous works in that it focusses on the use of sound examples. This has not been addressed before apart from [1], where we showed the effectiveness of Random Forest (RF) regression to model non-linear relationships between audio features and DRC control parameters in the context of isolated notes. Here, we adopt our framework to more complex audio material, mono-timbral *loops* that are commonly used by producers. Loops are short snippets of audio that can be repeated to create musical patterns. Many DAWs contain a loop library for building rich music layers. The design of appropriate features for loops is challenging since audio events may overlap. This makes direct measurement difficult, particularly during the attack and release phases of notes. We address this by testing different audio decomposition and feature extraction strategies that enable designing features relevant to controlling DRC parameters, particularly those related to ballistics, i.e., the attack and release times (τ_a, τ_r) of the DRC.

We choose three approaches to decompose loops. The most straightforward method is based on onset event detection. Guidelines for choosing the appropriate detection function are provided in [6]. Time domain methods are normally adequate for percussive signals, while spectral methods based on spectral or phase difference are suitable for pitched instruments. Complex-domain spectral difference works well but with higher computational cost while state-of-the-art methods using deep learning [7] are needed for polyphonic material. Since mono-timbral loops do not have such complex struc-

ture, we opt for the simple High Frequency Content (HFC) [6] detection function. The second approach is based on source separation using Non-negative Matrix Factorisation (NMF) to decompose complex audio into activation patterns [8]. Finally we assess transient/stationary audio separation. Researchers used orthogonal wavelet bases in [9] while others combined Modified Discrete Cosine Transform (MDCT) and wavelet bases [10]. We choose a more recent work using the Iterative Shrinkage Threshold Algorithm (ISTA) [11] for our purpose.

3. DECOMPOSITION AND FEATURE DESIGN

An overview of our method is shown in Fig.1. We aim at making the *Output* audio, generated from the first *Input* sound as close as possible to the second input *Reference*. Using a random forest regression model we map a vector of low level features related to each specific compressor parameter $\Theta = \{\tau_a, \tau_r, Ratio, Thd\}$. The key to good performance is designing or selecting the most relevant features.

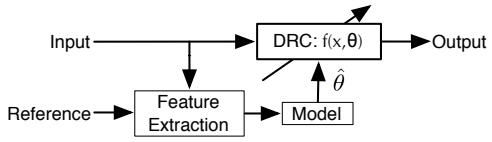


Fig. 1. System overview

We use standard audio features along with novel ones. Frame-wised spectral centroid, variance and RMS are extracted and the mean and variance across the frame are used as standard features due to their relation to dynamics. DRC involves several stages of non-linearity and has a different behaviour during transient and stationary parts of sounds due to the attack and release time parameters. We decompose loops into simpler audio excerpts so that the attack and release phases can be measured more accurately. Three approaches are applied and examined including onset event detection, NMF and transient/stationary separation using ISTA. The decomposition methods are applied before extracting features specific to attack time detailed in Eqn.1-3. The original design motivations are discussed in our previous work [1]. These correspond to the length, average energy and ascending speed of the attack phase. N_{startA} and N_{endA} represent the start and end positions of the attack phase, which is calculated using the RMS curve through a fixed threshold method (c.f. [12]). Release features are calculated in a similar manner.

$$\mathsf{T_A} = (N_{endA} - N_{startA})/Fs, \tag{1}$$

$$\mathsf{A1_{att}} = \frac{1}{N_{endA} - N_{startA}} \sum_{n=N_{startA}}^{N_{endA}} rms\_curve(n), \tag{2}$$

$$\mathsf{A2_{att}} = rms\_curve(N_{endA}), \tag{3}$$

3.1. Onset event detection

Using HFC we separate the onsets from a loop with the assumption that it contains a note between two onsets. We then apply feature extraction as described in Procedure 1. After obtaining onset positions, we choose notes with attack/release phases that have not been smeared by other notes. We choose these using two conditions. First we select notes that are longer than 1ms. Shorter notes normally indicate heavy overlap. The other condition is *goodness of fit* using two functions motivated by assumptions on note envelope. A polynomial function fitted on the ascending part of the note envelope and an exponential decay function on the descending part. If the fitted parameters do not show the ascending/descending trend the note is discarded. The procedure secures that only the clear attack/release phases within the loops are selected. Features are calculated according to Eqn.1-3 and averaged over selected notes. The parameters α and β in Procedure1 are the start window and the forward window size respectively. We forward the onsets to a forward window, in case the onsets does not appear at the beginning of the transient and afterwards we assume the start of the transient appears in the first 10% of the audio. We use a KZ filter [13] with 10 samples window size and 5 iterations to smooth the RMS curve before the process.

Procedure 1 Calculate designed features for mono loops.

Input:
$\mathcal{A}$ = Audio_Loop ; α = 10%; β = 0.2ms.
Output:
T_A; $A1_{att}$; $A2_{att}$.

K = OnsetEventDetection($\mathcal{A}$)
$k \in K$, K = *all the onset positions in the given loop*
for $k_i \in K$ **do**
  if $k_i - k_{i-1} < 1ms$ **then**
    skip;
  end if
  $k_i = k_i - \beta$
  R = RMS($\mathcal{A}[k_{i-1} : k_i]$)
  C = KZ_filter(R)
  s = argmin($C[0 : \alpha]$)
  p = argmax(C)
  e = size(C)
  $[a1, b1, c1]$=fit_poly($C[s : p]$)
  $[a2, b2, c2]$=fit_exp($C[p : e]$)
  if $(a1 > 0 \wedge -b1/2a1 > \beta) \vee (a1 < 0 \wedge -b1/2a1 < \beta) \vee (a_2/(e - p - b_2) > 0)$ **then**
    skip;
  end if
  $t_i = T_A(C)$; $a1_i = A1_{att}(C)$; $a2_i = A2_{att}(C)$
end for
$T_A = average(t)$; $A1_{att} = average(a1)$; $A2_{att} = average(a2)$

3.2. NMF

The second decomposition method uses spectral modelling, i.e., Non-negative matrix factorisation. NMF (c.f. Eqn.4) aims at decomposing the matrix **V** into a product of two non-negative matrices **W** and **H**. The target matrix **V** is the magnitude spectrogram of a the audio. In our case, it is the loop to

be decomposed. The spectrogram is generated using a window size of 4096 samples and an overlap of 1024 samples. The matrix $\mathbf{W}$ is the dictionary which contains C basis vectors. Meanwhile the matrix $\mathbf{H}$ is the activation pattern corresponding to each basis vector. In this approach, we use the activations $\mathbf{H}$ instead of the actual audio waveform to extract features. Since each row of $\mathbf{H}$ corresponds to a specific fraction of the loop, it is sparse and hence we can to retrieve the attack/release phases.

$$\mathbf{V}^{M\times N} \approx \mathbf{W}^{M\times C} * \mathbf{H}^{C\times N} \tag{4}$$

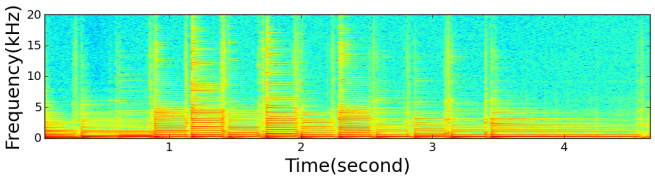
Unsupervised NMF suffers from a common limitation related to the dictionary recovery problem. Reasonable results can only be obtained for simple loops with only a small amount of non-overlapping notes. Without prior knowledge on the basis vectors, the activations may not correspond to events we wish to characterise.

To reduce the influence of this problem, semi-supervised NMF has been used. In the real world scenario, given a random loop, pre-trained dictionary based on the notes within this specific loop is not available. Therefore, we propose an alternative instrument specific method. Recent works on NMF based audio information retrieval methods are built upon fixed spectral templates representing harmonic components [14] or trained in an instrument specific manner [15]. Similarly, we use a set of twelve tone equal temperament acoustic guitar notes from RWC [16] library as the template set. This solution made the pre-trained dictionary sensitive to acoustic guitar timbre as well as the widely pitch range. We use 48 guitar notes across 3 octaves to form 4 such sets as training data for our dictionary, i.e. $\mathbf{w_i} \in [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_C}]$, $\mathbf{w_i} = \mathbf{1/4}\sum_{\mathbf{j=1}}^{\mathbf{j=4}} \mathbf{w_{ij}}$, with $C = 12$.

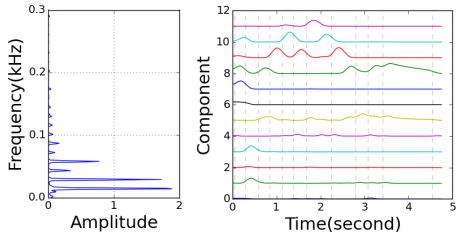
This dictionary is tested and verified on different acoustic guitar loops from the AppleLoop[1] library. An example of a loop which contains 13 notes is displayed. Its magnitude spectrogram is given in Fig.2(a). One dictionary element $\mathbf{w_{12}}$ from the fixed matrix $\mathbf{W}$ is given in Fig.2(b) which corresponds to the first activation pattern from the top in Fig.2(c). Although it is not possible to deliver perfect decomposition, it shows significant improvement over unsupervised NMF. Similar results are observed for other acoustic guitar loops.

The activation curves are examined to see if they have similar response with the actual energy curves when compressing the audio. The test shows positive results, since the activation curves are essentially the responses of individual notes. The activation patterns have a clear note-like shape and are sparse in general. As a result, we do not need to apply the complex selection strategies in Procedure 1, which makes this a more stable solution. We then calculate and average Eqn.1-3 from each activation and use the results as features.

[1] https://support.apple.com/kb/PH13426?locale=en_US&viewlocale=en_US



(a) Spectrogram

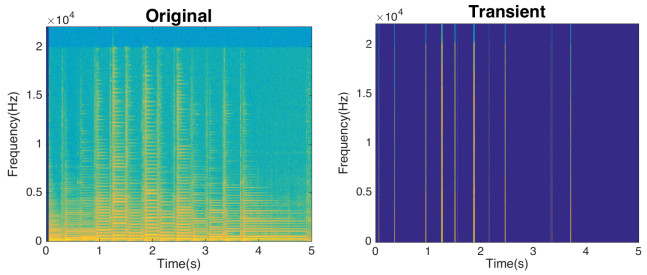


(b) Template No.12 (c) Activation pattern

Fig. 2. The spectrogram of an acoustic guitar loop (a), one of the fixed note template (b) and its decomposed activation pattern (c), using semi-supervised NMF.

3.3. Transient/Stationary audio separation

The final approach we propose is the decomposition of loops into transient and stationary (*T/S*) parts instead of individual notes. A state-of-the-art algorithm is proposed in [17] using Iterated Shrinkage/Threshold Algorithm framework, with a Matlab toolbox implementation[2]. An improvement over this using cross shrinking is proposed in [11] which provides good results for our case. An example of the separation is shown in Fig.3(a), 3(b). This algorithm is able to retrieve the start and stop positions of both transients and the stationary parts, which the transient positions can be used as N_{startA} and N_{endA} in Eqn.1-3 for attack features, and the stationary positions can be used for release features. We then compute features similarly to the previous cases.



(a) Spectrogram (b) Transient

Fig. 3. The spectrogram of an acoustic guitar loop and its transient positions, using ISTA.

4. EVALUATION AND RESULTS

There are two stages of evaluation. The first is a numerical test presented in Section 4.1. The predicted mean absolute error (MAE) of each parameter is reported when using each decomposition methods to extract features. The second is a sim-

[2] https://kaisiedenburg.net/research/

ilarity test using an audio similarity model discussed in Section 4.2. Audio materials for evaluation are 29 acoustic guitar, 30 electronic bass, and 12 drum loops from AppleLoop.

4.1. Numerical test

Since this paper focuses on attack time and release time, we generate the dataset by compressing N loops respectively to each parameter, (0,100]ms for attack time with step of 1ms, and (0,1000]ms for release time with step of 10ms. Therefore, we have $N*100$ compressed audio excerpts respectively. The model is aimed at learning the difference between the *Input* and the *Reference*, c.f. Fig.1. We form the training data by extracting features from each compressed audio and dividing them by the features extracted from the origins. The corresponding compressor parameters are used as training target for a random forest regression model. We use repeated random sub-sampling validation (Monte Carlo variation) to test the performance. 15% of each feature vectors are selected for testing, while the remaining are used as training. It is repeated 100 times and the average MAE is reported in Table 1. τ_a stands for attack time and τ_t for release time. *Std* stands for standard features, which represents the 6 high order statistical features c.f. [1]. The following labels, *Onset*, *NMF*, and *T/S*, represent the feature sets that contain both standard features and the ones generated using the labelled method.

For test cases, the error drops when adding the decomposition features onto standard features. NMF features provide the best performance comparing with the other individual features. However, using all features together produces the lowest error rate. Therefore, even though NMF stands out in this numerical evaluation, instead of choosing this specific feature, we use all three together for a better performance.

MAE(ms)	*Std*	*Onset*	*NMF*	*T/S*	*All*
$Guitar, \tau_a$	0.934	0.897	**0.845**	0.863	**0.807**
$Bass, \tau_a$	1.449	1.196	**1.071**	1.244	**0.995**
$Drum, \tau_a$	1.384	1.361	**1.194**	1.274	**1.134**
$Guitar, \tau_r$	12.115	10.604	**10.442**	11.802	**9.981**
$Bass, \tau_r$	11.701	11.143	**10.733**	10.886	**9.381**
$Drum, \tau_r$	16.327	14.946	**12.714**	13.315	**12.043**

Table 1. Predicted Mean Absolute Error(MAE) using different feature sets for three instrument loops.

4.2. Similarity test

In the previous section, we split the dataset for training and testing to evaluate the efficiency of the prediction model. In a more realistic situation, the *Reference* and the *Input* should be independent, c.f.Fig.1. In this section, we randomly selected 50 pairs of audio, using one as reference and the other one as origin. The model is able to give a predicted audio according to these two inputs. Therefore, we can evaluate the system by comparing $D1$ and $D2$, which represented the similarity distance between origin and reference, and prediction and reference respectively (c.f.Eqn.5). $D()$ represents the similarity distance measure function. Theoretically, the distance between prediction and reference should be smaller than the distance between input and reference.

$$\begin{aligned} D1 &= D(Iutput, Reference); \\ D2 &= D(Onput, Reference); \end{aligned} \tag{5}$$

We use a simple audio similarity model to test the efficiency of the system, which is also used in our earlier research[1]. MFCC coefficients are extracted and used to fit a Gaussian Mixture Model(GMM). An approximation of the symmetries KL divergence is then calculated and used as a distance measure. The average of 50 cases are displayed in Table2. Results show $D2$ are smaller than $D1$ for all cases, which means our method is able to bring the *Output* close to the *Reference* comparing with the *Input*. Since the actual value of the divergence does not have practical meaning, we normalised $D2$ according to $D1$, i.e. set $D1 = 1$, and only report the normalised results. $D2$ indicates the distance between the system output and the reference, therefore, the smallest is the best.

	$D2_{std}$	$D2_{onset}$	$D2_{nmf}$	$D2_{t/s}$	$D2_{all}$
$Guitar, \tau_a$	0.918	0.916	**0.914**	0.916	**0.916**
$Bass, \tau_a$	0.384	0.375	**0.371**	0.383	**0.362**
$Drum, \tau_a$	**0.251**	0.252	**0.251**	0.257	0.252
$Guitar, \tau_r$	0.934	0.936	0.940	**0.919**	**0.917**
$Bass, \tau_r$	0.738	0.732	**0.726**	0.733	**0.729**
$Drum, \tau_r$	0.583	0.589	**0.580**	**0.582**	0.584

Table 2. $D1$ and $D2$ comparison using different feature sets, when $D()$ is the audio perceptual similarity.

The trend from the numerical test is not consistent in the similarity test. We highlight the top two closest distance in each cases. NMF still distinguish to the other decomposition methods, however, the closest distance does not alway occur when using all three types decomposition methods. We found the average similarity are close for each case when we change the training feature sets. We then examined the prediction individually. The prediction parameter values are rather close ($<$1ms, c.f. Table 1), correspondingly the outputs of the similarity model are very close. It is reasonable because we use different decomposition method to extract same features, they are designed to provide similar information. The difference is their efficiency and complexity. Consider the results from both evaluation process, we can state that the most efficient decomposition method is NMF both numerically and perceptually, while using all three we will use all three sets of features together in the future work for better performance.

5. CONCLUSION AND FUTURE WORK

We explored three approaches of decomposing audio loops and extracting attack/release time related features. The use of these features have shown to be beneficial in our proposed framework for the intelligent control of the dynamic range compressor. The benefit is clear both in terms of the accuracy of predicting attack and release times as well as audio similarity using a simple perceptual model. Overall results show

that using all three feature sets works best numerically, while NMF stands out in both numerical and perceptual tests.

Our future work will be focussing on adapting our method to more complex audio materials, i.e., polyphonic tracks. The similarity model we used for evaluation is a widely used timbre similarity model. However we aim to develop models with a stronger focus on DRC parameters, as well as conducting listening tests for perceptual validation. We also aim to compare our approach with features extracted using deep learning techniques.

6. REFERENCES

[1] Di Sheng and György Fazekas, "Automatic control of the dynamic range compressor using a regression model and a reference sound," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

[2] Brecht De Man, Joshua D. Reiss, and Ryan Stables, "Ten years of automatic mixing," in *Proceedings of the 3rd Workshop on Intelligent Music Production*, 2017.

[3] Zheng Ma, Brecht De Man, Pedro DL Pestana, Dawn AA Black, and Joshua D Reiss, "Intelligent multitrack dynamic range compression," *Journal of the Audio Engineering Society*, vol. 63, no. 6, pp. 412–426, 2015.

[4] Mark Brozier Cartwright and Bryan Pardo, "Social-eq: Crowdsourcing an equalization descriptor map.," in *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.

[5] Philipp Kolhoff, Jacqueline Preub, and Jorn Loviscach, "Music icons: procedural glyphs for audio files," in *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2006, pp. 289–296.

[6] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on speech and audio processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[7] Jan Schluter and Sebastian Bock, "Improved musical onset detection with convolutional neural networks," in *Acoustics, speech and signal processing (ICASSP), 2014 ieee international conference on*. IEEE, 2014, pp. 6979–6983.

[8] Nancy Bertin, Roland Badeau, and Gaël Richard, "Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 1, pp. I–65.

[9] Gianpaolo Evangelista, "Pitch-synchronous wavelet representations of speech and music signals," *IEEE transactions on signal processing*, vol. 41, no. 12, pp. 3313–3330, 1993.

[10] Laurent Daudet and Bruno Torrésani, "Hybrid representations for audiophonic signal encoding," *Signal Processing*, vol. 82, no. 11, pp. 1595–1617, 2002.

[11] Kai Siedenburg and Simon Doclo, "Iterative structured shrinkage algorithms applied to stationary/transient separation," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

[12] Geoffroy Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," 2004.

[13] Wei Yang and Igor Zurbenko, "Kolmogorov–zurbenko filters," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 340–351, 2010.

[14] Nancy Bertin, Roland Badeau, and Emmanuel Vincent, "Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[15] Emmanouil Benetos, Sebastian Ewert, and Tillman Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3107–3111.

[16] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "Rwc music database: Music genre database and musical instrument sound database," *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pp. 229–230, 2003.

[17] Kai Siedenburg and Monika Dörfler, "Persistent time-frequency shrinkage for audio denoising," *Journal of the Audio Engineering Society*, vol. 61, no. 1/2, pp. 29–38, 2013.